PhageAcr: Identification of anti-CRISPR proteins

Christian Neitzel, Fernanda Vieira, Hugo Oliveira, Óscar Dias

Center of Biological Engineering, University of Minho, 4710-057 Braga, Portugal

Abstract. In the ongoing co-evolutionary arms race between bacteria and bacteriophages, the bacterial CRISPR-Cas system serves as a key defense mechanism, targeting viral DNA. Bacteriophages, in turn, have developed anti-CRISPR (Acr) proteins to inhibit these defenses. This study aims to enhance Acr protein identification by constructing a comprehensive database of Acr sequences and benchmarking various machine learning (ML) models, as well as sequence redundancy tools. Utilizing data from various Acr databases for positive samples, resulting in a dataset of 1,751 sequences, and phage structural protein data for negative samples, we removed sequence redundancies using CD-HIT and MMseqs2 tools. We then extracted physicochemical properties using Propythia and trained four ML models—Decision Tree, Random Forest, Support Vector Machines (SVM) and eXtreme Gradient Boosting (XGBoost)—to classify Acr proteins. Performance metrics indicated that Random Forest and XGBoost outperformed other models, with Random Forest achieving the highest Area Under Curve (AUC) of 0.980 on the CD-HIT dataset and XGBoost achieving 0.978 on the MMseqs2 dataset. Our findings emphasize the efficacy of ensemble learning methods in accurately identifying Acr proteins, providing a robust framework for future research in bacterial immune systems and phage interactions.

Keywords: anti-crispr · genetic engineering · machine learning · classification

1 Introduction

Bacteria and phages are locked in a co-evolutionary arms race, leading to the development of sophisticated defense mechanisms. One such defense in bacteria is the Clustered Regularly Interspaced Short Palindromic Repeats (CRISPR or CRISPR-Cas) system, which allows bacteria to precisely target and degrade viral DNA (Asmamaw & Zawdie, 2021; Hale et al., 2009; Horvath & Barrangou, 2010). This system consists of a CRISPR array and Cas (CRISPR-associated) genes (Makarova et al., 2015). The CRISPR array contains repeated sequences interspersed with variable-length "spacers", deriving from fragments of foreign DNA that were cleaved and modified into "protospacers" by Cas proteins. These protospacers integrate into the CRISPR array, augmenting the host's defense mechanisms against similar phages that may attempt future infections (Sternberg et al., 2016). Cas genes encompass a cluster of genes responsible for encoding Cas proteins, which are essential for CRISPR-Cas defense mechanisms and information processing within the CRISPR array (Alkhnbashi et al., 2020). To put it simply, Cas proteins identify, cleave and modify viral DNA, whereas the CRISPR array acts as a repository of the modified fragments of viral DNA (spacers), optimizing the bacteria's defenses against viral genomes (Horvath & Barrangou, 2010).

The CRISPR-Cas system, serves as an adaptive immune system, recognizing and degrading viral genetic material. Due to its precise genome editing capabilities, CRISPRs have gained significant attention across various areas, mainly in genetic engineering (Haq et al., 2012), pharmaceuticals (Guo et al., 2021) and agriculture (Liu et al., 2021). However, the manipulation of CRISPRs for gene editing and synthetic gene circuit construction necessitates safety measures to mitigate potential adverse effects (Choudhary et al., 2023; Hu et al., 2024).

In response to the CRISPR defense, certain phages have evolved anti-CRISPR (Acr) proteins, which inhibit the CRISPR-Cas systems, allowing successful phage invasion and replication (Bondy-Denomy et al., 2013). These Acr proteins consist of regulatory proteins that are injected into the bacterium during the phage infection, their purpose is to target a specific CRISPR-Cas system and inhibit the Cas protein tasked with cleaving the viral genome, neutralizing the bacterial defense mechanism.

Consequently, the evolution of these defense and countermeasure mechanisms would further stir the co-evolution between bacteria and phages, specially between CRISPRs and Acrs, forcing the former to increase in complexity and develop to become more specialized, leading to the appearance of a wide variety of alternate CRISPR systems (Makarova et al., 2015, 2020; Makarova & Koonin, 2015), and the latter to develop their variants to specifically counter these new developments (Koonin et al., 2017; Pawluk et al., 2018).

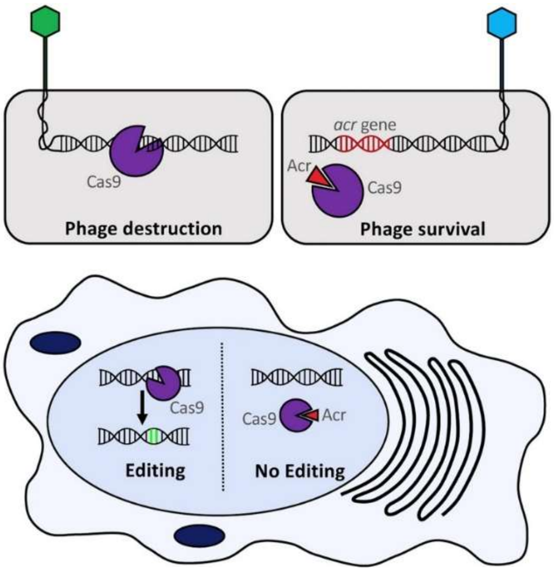

Fig. 1 – Comparison of CRISPR-Cas9 and anti-CRISPR mechanisms within a bacteria (Pawluk et al., 2016).

Anti-CRISPR Databases

The discovery of new Acr protein families makes it important to establish a standardized nomenclature to ensure clarity and consistency among researchers. Efforts towards this goal, as referenced in (Bondy-Denomy et al., 2015; Pawluk et al., 2016), have laid the groundwork for a unified Acr nomenclature. Building upon this foundation, researchers have established the Anti-CRISPR Assembly[1] (Bondy-Denomy et al., 2018), provides a comprehensive repository for cataloging newly identified Acr proteins, ensuring conformity to established naming conventions (Bondy-Denomy et al., 2018)

Two key databases for Acr research are Anti-CRISPRdb (Dong et al., 2018, 2022) and CRISPRminer (Zhang et al., 2018). Anti-CRISPRdb[2] extracts data from PubMed and Google Scholar, aligns sequences through NCBI's BLAST, and offers comprehensive functions for searching, browsing and downloading Acr data (Dong et al., 2018). CRISPRminer[3] on the other hand provides advanced analysis tools and information on Acr families, complementing Anti-CRISPRdb by offering insights into CRISPR-Cas systems.

Identification of Anti-CRISPRs

Identifying Acrs is challenging due to their variable amino acid sequences and compact size (Pawluk et al., 2018), which complicates traditional sequence analysis methods (Zhu et al., 2022). While conventional tools relied on guilt-by-association (GBA) and self-targeting methods, Machine Learning (ML) approaches can enhance Acr identification by training models on databases of confirmed Acrs and non-Acrs, learning distinguishing features that differentiate them (A. B. Gussow et al., 2020).

ML models of interest for this task include Decision Tree (DT), Random Forest (RF), Support Vector Machines (SVM) and eXtreme Gradient Boosting (XGBoost). DT is a simple tree structure where nodes represent features and branches represent decision rules, leading to outcomes based on those features (SONG & LU, 2015). The simplicity of DT allows for easy interpretation and visualization of decision-making processes, although they are prone to overfitting, especially with complex datasets (Srihith et al., 2023). RF is an ensemble of decision trees that improve predicting accuracy by constructing multiple trees during training and averaging their predictions (Breiman, 2001). RF generates diverse trees by training each tree on different random subsets of data and using random feature selection at each split, enhancing the model's robustness and generalizability (Wang, 2024). SVM classifies data by finding the optimal hyperplane that separates different classes (Cervantes et al., 2020). This model is versatile and effective in high-dimensional spaces due to their use of different kernel functions to handle various types of data and decision boundaries. SVM is powerful for binary classification tasks, becoming one of the most used classification methods (Cervantes et al., 2020). XGBoost is an optimized gradient boosting algorithm that iteratively improves model accuracy by minimizing prediction error (Chen & Guestrin, 2016). XGBoost includes advanced regularization techniques to prevent overfitting and uses a novel sparsity-aware algorithm to handle missing data efficiently. It also supports parallel processing, making it efficient and scalable for large datasets (Tarwidi et al., 2023).

[1] Available at https://tinyurl.com/anti-CRISPR

[2] Available at http://guolab.whu.edu.cn/anti-CRISPRdb/

[3] Available at http://www.microbiome-bigdata.com/CRISPRminer

Examples of Anti-CRISPR detection tools

The identification of Acr proteins has been significantly advanced by the development of various computational tools, each employing distinct methodologies to enhance detection accuracy and efficiency:

AcrFinder[4], integrates homology search, guilt-by-association (GBA) and CRISPR-Cas self-targeting spacer analysis to identify Acrs. This multifaceted approach increases the likelihood of accurate Acr identification (Yi et al., 2020).

AcRanker[5] (Eitzinger et al., 2020), utilizes the XGBoost algorithm (Chen & Guestrin, 2016), to rank proteins based on expected Acr behavior using amino acid compositions as input features. Trained on experimentally verified Acrs from the Anti-CRISPRdb database, AcRanker successfully identified new Acrs.

AcrCatalog[6] employs a RF algorithm (Breiman, 2001) to predict and characterize novel Acrs (A. Gussow et al., 2020). By enriching search spaces within prokaryotic and viral genomes and applying heuristic filters, AcrCatalog identified 2500 previously undetected Acr candidates, demonstrating robust predictive capabilities. This tool uses an ensemble of decision trees, each trained on random subsets of the training data, to make robust predictions (Breiman, 2001).

For this work we aimed to build a comprehensive database of Acr protein sequences, by leveraging from previously mentioned databases of curated Acrs. Additionally, we performed a benchmarking of different available tools and ML models to develop an accurate ML tool for predicting Acr protein sequences.

2 Methods

Leveraging from the Python Programming Language, we constructed a dataset with data taken from anti-CRISPRdb, CRISPRminer and Anti-CRISPR Assembly. This data provides our Acr sequences, which we define as our positive samples. By contrast, negative samples are defined as non-Acr sequences. This data is gathered from the Phage Artificial Neural Networks (PhANNs) database (Cantu et al., 2020). Our ML model was built to make binary classifications, as such, we constructed a dataset that was fed into the algorithm so that it may be trained upon. This dataset consisted of a positive class, which contains our known Acr protein sequences, and a negative class, which consists of non-Acr protein sequences.

Positive Dataset Construction

To construct our positive dataset, we sourced data from Anti-CRISPRdb (Dong et al., 2022), CRISPRminer (Zhang et al., 2018) and the Anti-CRISPR Assembly (Bondy-Denomy et al., 2018). This data was preprocessed using functions from the Pandas package in Python. Each database was analyzed to identify and clean up potential inconsistencies and errors, such as invalid characters or missing sequences.

[4] Available at http://bcb.unl.edu/AcrFinder

[5] Available at https://bio.tools/AcRanker

[6] Available at https://acrcatalog.pythonanywhere.com/catalog/

For the Anti-CRISPRdb database, we downloaded both the XLS and CSV files of the core dataset from their website, preprocessed both these files and discovered that between them 44 sequences were not present in both the datasets. Upon concatenating them and performing further preprocessing we obtained a dataset of 1,715 sequences from Anti-CRISPRdb. The dataset from CRISPRminer required minimal preprocessing, mainly renaming columns and cleaning content using regular expressions, resulting in 21 sequences. When compared to the preprocessed Anti-CRISPRdb dataset, we discovered that all accession IDs and sequences in CRISPRminer were already present in Anti-CRISPRdb but while certain sequences were the same, their accession IDs were different and vice-versa. Anti-CRISPR Assembly involved extensive preprocessing due to sequences being spread across multiple Excel sheets without their accession IDs. Preprocessing this dataset and comparing it with our preprocessed Anti-CRISPRdb and CRISPRminer datasets, we were able to pinpoint 40 sequences that were unique to Anti-CRISPR Assembly. We then used BLAST queries by leveraging from the Bio.Blast subpackage of Biopython to access the NCBI database. This way, we were able to retrieve the missing accession IDs. Combining all processed data and performing final treatment of our data, we constructed a positive dataset of 1,751 sequences, which was converted into FASTA format.

Negative Dataset Construction

For the negative dataset, we used the Phage Artificial Neural Networks[7] (PhANNs) database (Cantu et al., 2020), which categorizes phage protein sequences into structural proteins and regulatory proteins. The selection of PhANNs for our study was strategic. When searching for Acrs, it is logical to search within the realm of phage proteins to ensure the context and environment of the search are appropriate for the target sequences.

PhANNs separates phage protein sequences by their functions. Specifically, it distinguishes structural protein from regulatory proteins. Since Acrs are classified as regulatory, we selected structural proteins for our negative dataset. This approach minimizes the risk of including potential Acr proteins, as structural proteins are unlikely to be misclassified as Acrs.

By using the PhANNs database, we ensured that our negative dataset consisted of phage-associated proteins while avoiding the inclusion of Acrs. This selection provided a set of 168,660 sequences, which we then converted into FASTA format.

Sequence Redundancy Removal

Upon completion of the positive and negative datasets, we aimed to simplify our data by removing redundant sequences using two sequence redundancy removal tools: Cluster Database at High Identity with Tolerance (CD-HIT) (Li & Godzik, 2006) and Many-against-Many Sequence Searching (MMseqs2) (Steinegger & Söding, 2017).

The CD-HIT tool clusters sequences based on their similarity, retaining one representative per cluster (Li & Godzik, 2006). To increase the sensitivity of the clustering algorithm we set a word size of 5. The MMseqs2 tool also clusters sequences based on similarity but offers enhanced speed and scalability compared to CD-HIT (Steinegger & Söding, 2017).

[7] Available at http://phanns.com/

For both tools, we defined a threshold of 90% sequence similarity for the sequence clustering algorithms. Both tools input and output FASTA format files, ensuring compatibility and ease of use in our workflow. By employing these tools, we reduced redundancy in our datasets, ensuring that our subsequent analyses were based on a simplified and representative set of sequences.

The output that CD-HIT provided was 1,122 sequences for the positive dataset and 57,379 sequences for the negative dataset. As for MMseqs2, the output was 1,106 sequences for the positive dataset and 58,286 sequences for the negative dataset. Since we want to have the same number of positive and negative sequences, we randomly selected for the negative datasets of both CD-HIT and MMseqs2, the same number of sequences as their positive dataset counterparts, producing negative datasets of 1,122 sequences and 1,106 sequences respectively.

Sequence Feature Acquisition

We used Propythia (Sequeira et al., 2022), a Python-based tool designed for extracting sequence features. This tool enabled the conversion of sequence data into descriptors, facilitating compatibility with our ML algorithms. Some of these functions can be as specific as just obtaining the physicochemical features, molecular bond composition, amino acid compositions, length, etc. Using this tool we performed the function "get_physicochemical", which allowed us to obtain the physicochemical properties of our sequences, providing essential features for ML algorithms.

Upon extracting the features of non-redundant positive and negative datasets, we joined their respective pairs together, removed non-numerical columns and set the accession IDs as the concatenated datasets' index column, producing a CD-HIT dataset with 2,244 sequences and an MMseqs2 dataset with 2,212 sequences.

Machine Learning

For the ML, we leveraged on the scikit-learn package (Pedregosa et al., 2012), we also intend on benchmarking different widely used models on non-redundant outputs from CD-HIT and MMseqs2. All ML steps done on the CD-HIT dataset were repeated for the MMseqs2 dataset.

The dataset was randomly divided into three sets: 70% for training, 20% for testing and 10% for validation. The training set consists of data that our models will train from, and the test set to evaluate model performance. The validation set, consisting of unseen data, will be used to verify whether overfitting occurs, thereby validating the efficacy of the final model in realistic deployment.

We selected four ML models for benchmarking: DT, RF, SVM and XGBoost. Each model underwent hyperparameter optimization using scikit-learn's *GridSearchCV* function, which performed an exhaustive search over specified parameter values to find the best-scoring estimator.

3 Results and Discussion

The performance of the selected models was evaluated using the metrics for accuracy, precision, recall or sensitivity, specificity and F1-Score, as we can see in Table 1.

Table 1 – Metric scores of each ML model using their best hyperparameters, evaluated on CD-HIT and MMseqs2 non-redundant datasets

	CD-HIT				MMseqs2			
	DT	RF	SVM	XGB	DT	RF	SVM	XGB
Accuracy	0.88	0.93	0.92	0.92	0.88	0.93	0.92	0.93
Precision	0.85	0.90	0.91	0.89	0.90	0.91	0.91	0.92
Recall	0.91	0.97	0.94	0.96	0.87	0.95	0.94	0.94
Specificity	0.84	0.90	0.90	0.88	0.90	0.90	0.90	0.92
F1-Score	0.88	0.94	0.92	0.93	0.88	0.93	0.92	0.93

From the obtained metrics, we can observe that all models, regardless of the tool used for sequence redundancy, showed high predictive power. RF and XGBoost outperformed in terms of accuracy. All models scored above 90% precision save for the DT model trained on the CD-HIT dataset, indicating strong ability to correctly identify Acr proteins. For the recall metric, RF, SVM and XGBoost performed best, reflecting their effectiveness in identifying most of the actual Acr sequences. In terms of specificity, the models performed overall better on the MMseqs2 dataset, demonstrating strong performance in correctly identifying non-Acr sequences. The F1-Score, which evaluates the balance between precision and recall, was the highest for RF, SVM and XGBoost across both datasets, indicating balanced and robust classification performance.

RF excelled in recall across both datasets, indicating its strong capability to identify actual Acr sequences, and showed the highest or near-highest performance in all metrics, suggesting it is a highly reliable model. XGBoost displays excellent balanced performance with high F1-Scores, making it a highly effective classifier. Its consistently high precision and specificity highlight its effectiveness in identifying non-Acr sequences. SVM performs well in precision and recall, especially on the CD-HIT dataset and its high specificity and F1-Scores indicated balanced classification performance. While the DT model was effective, it had comparatively lower performance, particularly in recall and specificity, making it less reliable than ensemble methods and SVM. Overall, RF and XGBoost models exhibited superior performance across both datasets, making them the recommended classifiers for distinguishing Acr and non-Acr protein sequences.

To evaluate the potential for overfitting in our ML models, we monitored their performance on the validation set after the training phase. The validation set accuracy scores provide insights into how well the models generalize to unseen data (Table 2).

Table 2 – Validation set accuracy scores of each ML model used on CD-HIT and MMseqs2 non-redundant datasets

	CD-HIT				MMseqs2			
	DT	RF	SVM	XGB	DT	RF	SVM	XGB
Validation Set Accuracy	0.84	0.89	0.91	0.91	0.79	0.90	0.92	0.91

The DT model showed lower validation accuracy compared to other models, with scores of 0.84 on the CD-HIT dataset and 0.79 on the MMseqs2 dataset. These relatively lower scores suggest that the DT model might be more prone to overfitting, especially on the MMseqs2 dataset. The RF model performed consistently well on both datasets, achieving validation accuracies of 0.89 on CD-HIT and 0.90 on MMseqs2. These high and consistent scores across both datasets indicate good generalization ability and low risk of overfitting. The SVM model demonstrated excellent performance, with validation accuracies of 0.91 on the CD-HIT dataset and 0.92 on the MMseqs2 dataset. This consistency suggests that the SVM model is well-tuned and generalizes effectively to unseen data. The XGBoost model also showed strong performance, with validation accuracies of 0.91 on both CD-HIT and MMseqs2 datasets. The model's high accuracy on both datasets indicates robust performance and minimal overfitting.

The validation set accuracy scores provide a clear indication of each model's generalization capabilities. RF, SVM, and XGBoost models exhibited high and consistent validation accuracies, suggesting strong generalization and low risk of overfitting. In contrast, the DT model's lower accuracy, particularly on the MMseqs2 dataset, points towards a higher likelihood of overfitting. RF and XGBoost emerged as the most reliable models for predicting Acrs, followed closely by SVM.

To evaluate the performance of our classification models, we plotted their respective Receiver Operator Characteristic (ROC) curves and calculated the Area Under Curve (AUC) of each model. The ROC curve represents the diagnostic ability of our binary classifier system as its discrimination threshold is varied, while the AUC provides a single measure of the models' ability to distinguish between the positive and negative classes, with a higher AUC indicating better overall performance (Fig. 1).

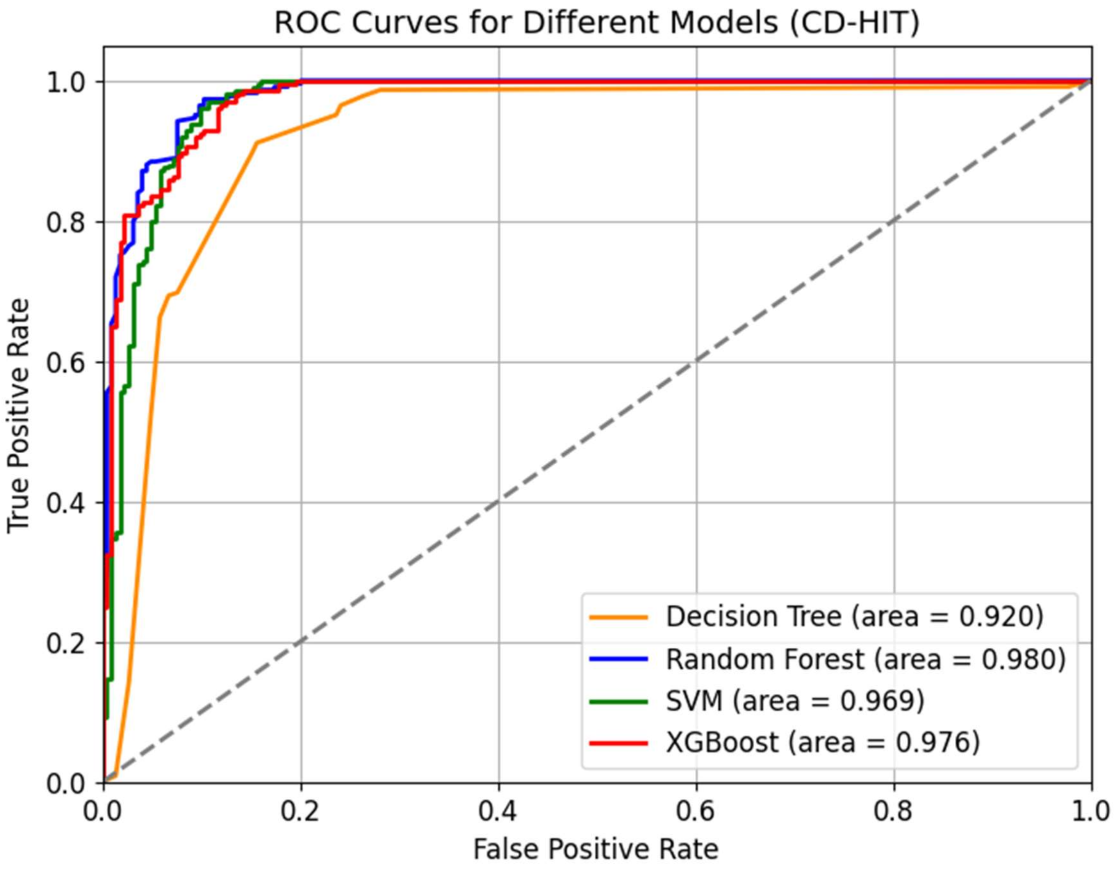

Fig. 2 – Receiver Operator Characteristic (ROC) Curves and Area Under Curve (AUC) of the models trained on the CD-HIT output dataset.

The ROC curves of all models suggest good performance in distinguishing between Acr and non-Acr sequences. The DT model scored the lowest for the CD-HIT dataset with an AUC of 0.920, while the RF model achieved the highest AUC at 0.980. This demonstrates that all four models possess high predictive power, effectively making the correct classifications while minimizing false positives and false negatives.

Next, we compare the ROC curve and AUC results of CD-HIT with those from the MMseqs2 dataset (Fig. 2).

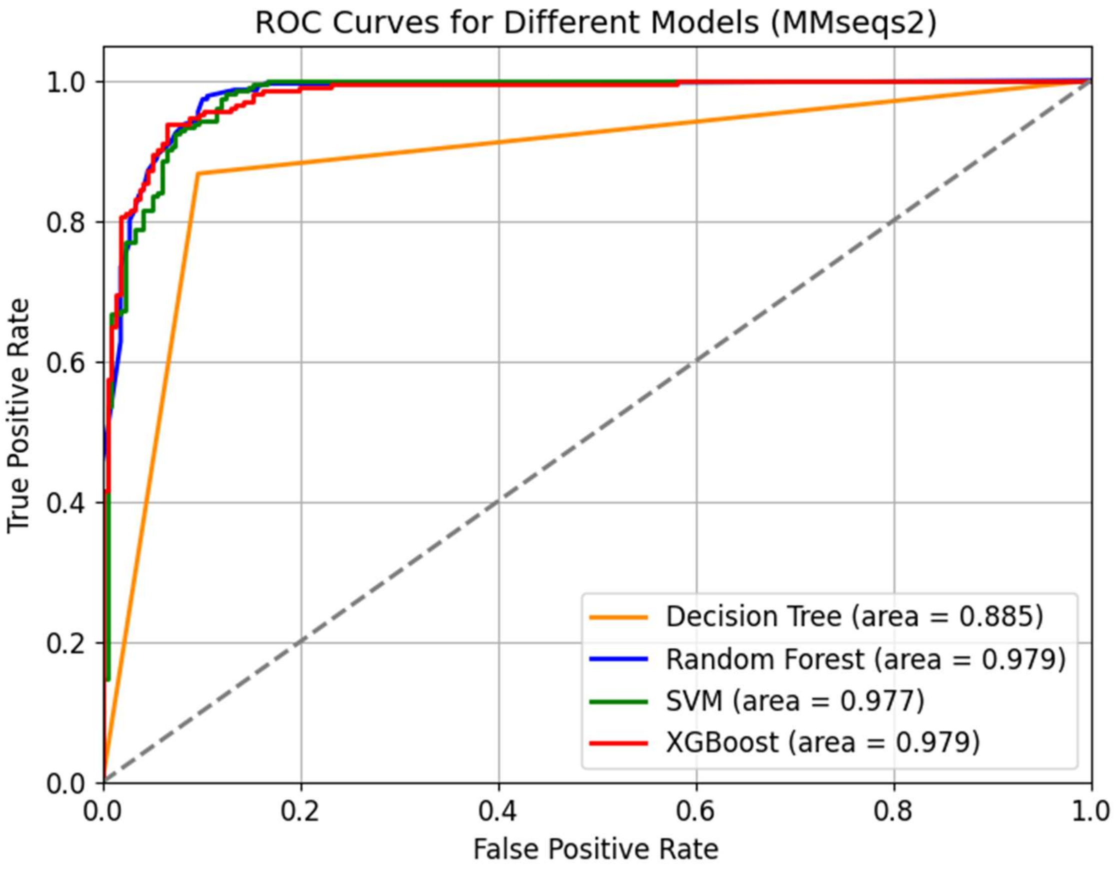

Fig. 3 – Receiver Operator Characteristic Curves (ROC) and Area Under Curve (AUC) of the models trained on MMseqs2 output dataset.

In this case, the DT model shows a lower AUC compared not only to the other models trained on the MMseqs2 dataset, but also to its CD-HIT counterpart. This is evident from its ROC curve which has a sharper elbow section compared to the other models' ROC curves. This behavior is typical of the DT's nature of making binary, hard decisions based on threshold splits at each node. It indicates that this DT model can achieve high sensitivity (true positive rate) with a minimal increase in false positive rate up to a certain point, after which any further increase in sensitivity comes at a higher cost of increasing the false positive rate. This leads to a potentially less generalizable, more overfitted behavior compared to smoother models.

As for the remaining models, all demonstrated good performance, indicating a strong ability to discriminate between Acr and non-Acr sequences. Summarizing the results of all our models on both CD-HIT and MMseqs2 datasets, we can rank them by AUC score.

Table 3 – Ranking of all models in CD-HIT and MMseqs2 output datasets by AUC score.

AUC Rank	Model	Sequence Clustering Tool	AUC score
1	Random Forest	CD-HIT	0.980
2	XGBoost	MMseqs2	0.979
3	Random Forest	MMseqs2	0.979
4	SVM	MMseqs2	0.977
5	XGBoost	CD-HIT	0.976
6	SVM	CD-HIT	0.966
7	Decision Tree	CD-HIT	0.920
8	Decision Tree	MMseqs2	0.885

These results indicate that while all models perform well, the RF and XGBoost models consistently exhibit superior performance across both datasets, making them the recommended classifiers for distinguishing between Acr and non-Acr protein sequences.

4 Conclusion

The comparative analysis of four ML models—DT, RF, SVM and XGBoost—revealed that RF and XGBoost consistently outperformed the other models across various performance metrics, including accuracy, precision, recall, specificity and F1-Score. The results were robust across both CD-HIT and MMseqs2 datasets, indicating the reliability and generalization of these models for Acr identification.

The ROC curves and the AUC metrics further substantiated the superior performance of RF and XGBoost models. RF achieved the highest AUC of 0.980 on the CD-HIT dataset, while XGBoost closely followed with an AUC of 0.979 on the MMseqs2 dataset. These findings highlight the effectiveness of ensemble learning methods in accurately distinguishing between Acr and non-Acr protein sequences.

Overall, our study showcases the importance of rigorous dataset preparation and the utility of ensemble ML models in bioinformatics applications. The RF and XGBoost models demonstrated good predictive capabilities, making them the recommended tools for future research in Acr protein identification. This works contributes to the growing body of knowledge in bacterial immune systems and phage interactions, providing robust framework for further advancements in the field.

Future improvements can be achieved through various enhancements and optimizations. Refining the acquisition of non-redundant, by experimenting with different parameter settings, could lead to more comprehensive datasets. Implementing advanced feature extraction methods would capture more complex patterns within sequences. Additionally, integrating extra data sources and databases could enhance the diversity of the training data. Experimenting with different ratios of positive to negative samples could help evaluate their impact on model performance. Increasing prediction complexity, by developing capabilities to identify and classify Acrs by their specific family and type, would further enhance the models. Benchmarking these models against existing Acr identification tools would validate performance and highlight areas for improvement, with a focus on incorporating a wider range of performance metrics for comprehensive assessment. Incorporating state-of-the-art language models can leverage their sequence analysis capabilities for feature extraction and classification, thereby improving model performance. Addressing these areas can significantly enhance the identification and classification of Acr proteins, contributing to more precise and effective research in Acr prediction and classification.

References

Alkhnbashi, O. S., Meier, T., Mitrofanov, A., Backofen, R., & Voß, B. (2020). CRISPR-Cas bioinformatics. *Methods*, *172*, 3–11. https://doi.org/10.1016/j.ymeth.2019.07.013

Asmamaw, M., & Zawdie, B. (2021). Mechanism and Applications of CRISPR/Cas-9-Mediated Genome Editing. *Biologics : Targets & Therapy*, *15*, 353–361. https://doi.org/10.2147/BTT.S326422

Bondy-Denomy, J., Davidson, A. R., Doudna, J. A., Fineran, P. C., Maxwell, K. L., Moineau, S., Peng, X., Sontheimer, E. J., & Wiedenheft, B. (2018). A Unified Resource for Tracking Anti-CRISPR Names. *The CRISPR Journal*, *1*(5), 304–305. https://doi.org/10.1089/crispr.2018.0043

Bondy-Denomy, J., Garcia, B., Strum, S., Du, M., Rollins, M. F., Hidalgo-Reyes, Y., Wiedenheft, B., Maxwell, K. L., & Davidson, A. R. (2015). Multiple mechanisms for CRISPR-Cas inhibition by anti-CRISPR proteins. *Nature*, *526*(7571), 136–139. https://doi.org/10.1038/nature15254

Bondy-Denomy, J., Pawluk, A., Maxwell, K. L., & Davidson, A. R. (2013). Bacteriophage genes that inactivate the CRISPR/Cas bacterial immune system. *Nature*, *493*(7432), 429–432. https://doi.org/10.1038/nature11723

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Cantu, V. A., Salamon, P., Seguritan, V., Redfield, J., Salamon, D., Edwards, R. A., & Segall, A. M. (2020). PhANNs, a fast and accurate tool and web server to classify phage structural proteins. *PLoS Computational Biology*, *16*(11), e1007845. https://doi.org/10.1371/journal.pcbi.1007845

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, *408*, 189–215. https://doi.org/10.1016/j.neucom.2019.10.118

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Choudhary, N., Tandi, D., Verma, R. K., Yadav, V. K., Dhingra, N., Ghosh, T., Choudhary, M., Gaur, R. K., Abdellatif, M. H., Gacem, A., Eltayeb, L. B., Alqahtani, M. S., Yadav, K. K., & Jeon, B.-H. (2023). A comprehensive appraisal of mechanism of anti-CRISPR proteins: An advanced genome editor to amend the CRISPR gene editing. *Frontiers in Plant Science*, *14*, 1164461. https://doi.org/10.3389/fpls.2023.1164461

Dong, C., Hao, G.-F., Hua, H.-L., Liu, S., Labena, A. A., Chai, G., Huang, J., Rao, N., & Guo, F.-B. (2018). Anti-CRISPRdb: A comprehensive online resource for anti-CRISPR proteins. *Nucleic Acids Research*, *46*(Database issue), D393–D398. https://doi.org/10.1093/nar/gkx835

Dong, C., Wang, X., Ma, C., Zeng, Z., Pu, D.-K., Liu, S., Wu, C.-S., Chen, S., Deng, Z., & Guo, F.-B. (2022). Anti-CRISPRdb v2.2: An online repository of anti-CRISPR proteins including information on inhibitory mechanisms, activities and neighbors of curated anti-CRISPR proteins. *Database*, *2022*, baac010. https://doi.org/10.1093/database/baac010

Eitzinger, S., Asif, A., Watters, K. E., Iavarone, A. T., Knott, G. J., Doudna, J. A., & Minhas, F. ul A. A. (2020). Machine learning predicts new anti-CRISPR proteins. *Nucleic Acids Research*, *48*(9), 4698–4708. https://doi.org/10.1093/nar/gkaa219

Guo, D., Chen, J., Zhao, X., Luo, Y., Jin, M., Fan, F., Park, C., Yang, X., Sun, C., Yan, J., Chen, W., & Liu, Z. (2021). Genetic and Chemical Engineering of Phages for Controlling Multidrug-Resistant Bacteria. *Antibiotics*, *10*(2), 202. https://doi.org/10.3390/antibiotics10020202

Gussow, A. B., Park, A. E., Borges, A. L., Shmakov, S. A., Makarova, K. S., Wolf, Y. I., Bondy-Denomy, J., & Koonin, E. V. (2020). Machine-learning approach expands the repertoire of anti-CRISPR protein families. *Nature Communications*, *11*(1), 3784. https://doi.org/10.1038/s41467-020-17652-0

Gussow, A., Shmakov, S., Makarova, K., Wolf, Y., Bondy-Denomy, J., & Koonin, E. (2020). *Vast diversity of anti-CRISPR proteins predicted with a machine-learning approach*. https://doi.org/10.1101/2020.01.23.916767

Hale, C. R., Zhao, P., Olson, S., Duff, M. O., Graveley, B. R., Wells, L., Terns, R. M., & Terns, M. P. (2009). RNA-Guided RNA Cleavage by a CRISPR RNA-Cas Protein Complex. *Cell*, *139*(5), 945–956. https://doi.org/10.1016/j.cell.2009.07.040

Haq, I. U., Chaudhry, W. N., Akhtar, M. N., Andleeb, S., & Qadri, I. (2012). Bacteriophages and their implications on future biotechnology: A review. *Virology Journal*, *9*, 9. https://doi.org/10.1186/1743-422X-9-9

Horvath, P., & Barrangou, R. (2010). CRISPR/Cas, the immune system of bacteria and archaea. *Science (New York, N.Y.)*, *327*(5962), 167–170. https://doi.org/10.1126/science.1179555

Hu, C., Myers, M. T., Zhou, X., Hou, Z., Lozen, M. L., Nam, K. H., Zhang, Y., & Ke, A. (2024). Exploiting activation and inactivation mechanisms in type I-C CRISPR-Cas3 for genome-editing applications. *Molecular Cell*, *84*(3), 463-475.e5. https://doi.org/10.1016/j.molcel.2023.12.034

Koonin, E. V., Makarova, K. S., & Zhang, F. (2017). Diversity, classification and evolution of CRISPR-Cas systems. *Current Opinion in Microbiology*, *37*, 67–78. https://doi.org/10.1016/j.mib.2017.05.008

Li, W., & Godzik, A. (2006). Cd-Hit: A Fast Program for Clustering and Comparing Large Sets of Protein or Nucleotide Sequences. *Bioinformatics (Oxford, England)*, *22*, 1658–1659. https://doi.org/10.1093/bioinformatics/btl158

Liu, Q., Yang, F., Zhang, J., Liu, H., Rahman, S., Islam, S., Ma, W., & She, M. (2021). Application of CRISPR/Cas9 in Crop Quality Improvement. *International Journal of Molecular Sciences*, *22*(8), 4206. https://doi.org/10.3390/ijms22084206

Makarova, K. S., & Koonin, E. V. (2015). Annotation and Classification of CRISPR-Cas Systems. *Methods in Molecular Biology (Clifton, N.J.)*, *1311*, 47–75. https://doi.org/10.1007/978-1-4939-2687-9_4

Makarova, K. S., Wolf, Y. I., Alkhnbashi, O. S., Costa, F., Shah, S. A., Saunders, S. J., Barrangou, R., Brouns, S. J. J., Charpentier, E., Haft, D. H., Horvath, P., Moineau, S., Mojica, F. J. M., Terns, R. M., Terns, M. P., White, M. F., Yakunin, A. F., Garrett, R. A., van der Oost, J., … Koonin, E. V. (2015). An updated evolutionary classification of CRISPR–Cas systems. *Nature Reviews Microbiology*, *13*(11), 722–736. https://doi.org/10.1038/nrmicro3569

Makarova, K. S., Wolf, Y. I., Iranzo, J., Shmakov, S. A., Alkhnbashi, O. S., Brouns, S. J. J., Charpentier, E., Cheng, D., Haft, D. H., Horvath, P., Moineau, S., Mojica, F. J. M., Scott, D., Shah, S. A., Siksnys, V., Terns, M. P., Venclovas, Č., White, M. F., Yakunin, A. F., … Koonin, E. V. (2020). Evolutionary classification of CRISPR-Cas systems: A burst of class 2 and derived variants. *Nature Reviews. Microbiology*, *18*(2), 67–83. https://doi.org/10.1038/s41579-019-0299-x

Pawluk, A., Amrani, N., Zhang, Y., Garcia, B., Hidalgo-Reyes, Y., Lee, J., Edraki, A., Shah, M., Sontheimer, E. J., Maxwell, K. L., & Davidson, A. R. (2016). Naturally Occurring Off-Switches for CRISPR-Cas9. *Cell*, *167*(7), 1829-1838.e9. https://doi.org/10.1016/j.cell.2016.11.017

Pawluk, A., Davidson, A. R., & Maxwell, K. L. (2018). Anti-CRISPR: Discovery, mechanism and function. *Nature Reviews Microbiology*, *16*(1), 12–17. https://doi.org/10.1038/nrmicro.2017.120

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., & Louppe, G. (2012). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*.

Sequeira, A. M., Lousa, D., & Rocha, M. (2022). ProPythia: A Python package for protein classification based on machine and deep learning. *Neurocomputing*, *484*, 172–182. https://doi.org/10.1016/j.neucom.2021.07.102

SONG, Y., & LU, Y. (2015). Decision tree methods: Applications for classification and prediction. *Shanghai Archives of Psychiatry*, *27*(2), 130–135. https://doi.org/10.11919/j.issn.1002-0829.215044

Srihith, I. V., Lakshmi, P., Donald, A., Aditya, T., Srinivas, T. A., & Thippanna, G. (2023). *A Forest of Possibilities: Decision Trees and Beyond*. *6*, 1–9. https://doi.org/10.5281/zenodo.8372196

Steinegger, M., & Söding, J. (2017). MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, *35*(11), 1026–1028. https://doi.org/10.1038/nbt.3988

Sternberg, S. H., Richter, H., Charpentier, E., & Qimron, U. (2016). Adaptation in CRISPR-Cas Systems. *Molecular Cell*, *61*(6), 797–808. https://doi.org/10.1016/j.molcel.2016.01.030

Tarwidi, D., Pudjaprasetya, S. R., Adytia, D., & Apri, M. (2023). An optimized XGBoost-based machine learning method for predicting wave run-up on a sloping beach. *MethodsX*, *10*, 102119. https://doi.org/10.1016/j.mex.2023.102119

Wang, T. (2024). Improved random forest classification model combined with C5.0 algorithm for vegetation feature analysis in non-agricultural environments. *Scientific Reports*, *14*(1), 10367. https://doi.org/10.1038/s41598-024-60066-x

Yi, H., Huang, L., Yang, B., Gomez, J., Zhang, H., & Yin, Y. (2020). AcrFinder: Genome mining anti-CRISPR operons in prokaryotes and their viruses. *Nucleic Acids Research*, *48*(W1), W358–W365. https://doi.org/10.1093/nar/gkaa351

Zhang, F., Zhao, S., Ren, C., Zhu, Y., Zhou, H., Lai, Y., Zhou, F., Jia, Y., Zheng, K., & Huang, Z. (2018). CRISPRminer is a knowledge base for exploring CRISPR-Cas systems in microbe and phage interactions. *Communications Biology*, *1*, 180. https://doi.org/10.1038/s42003-018-0184-6

Zhu, L., Wang, X., Li, F., & Song, J. (2022). PreAcrs: A machine learning framework for identifying anti-CRISPR proteins. *BMC Bioinformatics*, *23*(1), 444. https://doi.org/10.1186/s12859-022-04986-3